Prediction of USD Exchange Rate Against CNY and RUB Using Support Vector Regression and Neural Network

M. Fariz Fadillah Mardianto[1*], Larisa Mutiara Putri[2], Evi Wijayawati[3], and Sugha Faiz Al Maula[4]
[1,2,3,4]Author Affiliation: Department Math, Airlangga University, Surabaya, Indonesia
*Corresponding author: m.fariz.fadillah.m@fst.unair.ac.id

Received: 16 March 2025
Revised: 20 May 2026
Accepted: 25 May 2026

ABSTRACT – **Major currency exchange rates have been impacted by the escalation of global trade volatility brought on by the trade war between the United States and China and economic sanctions imposed on Russia. USD dominance in global trade exposes developing countries to economic risks. BRICS seeks to reduce reliance by boosting local currency trade and diversifying reserves. This study analyzes BRICS exchange rate movements, specifically USD-RUB and USD-CNY, using Support Vector Regression (SVR) and Neural Network (NN). Statistical analysis of 2009-2025 data shows USD-RUB's high volatility due to oil prices and sanctions, while USD-CNY remains more stable but is influenced by monetary policy and global conditions. The results show that the SVR method is superior to NN in prediction accuracy. For USD-RUB, SVR with a sigmoid kernel achieves MSE 6.1596, MAE 1.8808, and MAPE 1.95%, while for USD-CNY, SVR with a Radial Basis Function kernel achieves MSE 0.0014, MAE 0.0322, and MAPE 0.45% Thus, the use of SVR-based prediction models is recommended to analyze the exchange rate to reduce the risk of volatility. Additionally, diversifying reserves, enhancing bilateral trade in local currencies, and considering external factors like commodity prices and global policies can improve exchange rate stability and economic resilience.**

Keywords – CNY, RUB, Support Vector Regression, Neural Network, Exchange Rate

I. INTRODUCTION

One of the impacts of dependence on the United States (US) dollar is vulnerability to fluctuations in currency exchange rates. To address this, BRICS member countries are working to reduce the use of the dollar in bilateral and multilateral trade and seek other alternatives to protect their countries' economic interests [1]. BRICS exists to meet the global economic needs of developing countries and create a balance of power in the world economy [2]. BRICS' efforts to reduce dependence on the US dollar aim to build a multicurrency-based trading system and foreign exchange reserves so that the currencies of BRICS member countries are not easily affected or devalued when global economic turmoil occurs. In 2020, Russia pursued a de-dollarization policy by reducing the use of dollars in its trade balance, foreign debt, and banking assets. In addition, some large Russian energy companies have begun transacting using rubles. Secondly, Russia, China, and India have signed bilateral trade agreements using their respective national currencies, replacing the dollar's role in economic transactions [2].

Since 2018, trade tensions between the US and China have intensified, causing both countries to impose tariffs on each other. This trade war has negatively impacted both trade flows, with China experiencing a more significant impact [3]. China and the United States' trade conflict heated up again after China announced additional tariffs of up to 15% on some US goods, mainly agricultural products such as corn and soybeans, and export restrictions to 15 US companies starting March 10, 2025. The move is in retaliation to new tariffs imposed by the US on Chinese goods, which now total 20%. Beijing condemned Washington's policy, calling it damaging to trade relations and urging the US to lift the tariffs. In response, China is also considering further retaliation against US agricultural products, which are one of the main exports to the country [4]. In addition, Russian President Vladimir Putin commented on the potential for a volume II trade war under the leadership of US President Donald Trump. Putin predicted that countries on the continent would soon bow down and follow US directives in economic and trade policy [5]. On the other hand, the Kremlin emphasized Russia's stance against a global trade war due to its negative impact on all countries. Kremlin Spokesperson Dmitry Peskov highlighted the US policy to impose 25 percent tariffs on goods from the European Union, which is considered to worsen the world trade situation [6]. The high tariffs and economic sanctions imposed by the US on China and Russia have destabilized global trade. As a result, one major worry is the volatility of currency exchange rates, especially the movement of the Chinese yuan and the Russian ruble against the US dollar. Consequently, precise evaluation and forecasting of the ruble and yuan exchange rates in relation to the USD are necessary.

One of the data analysis methods that can be used to predict a currency of the nation exchange rate is the Support Vector Regression (SVR) method. This algorithm produces output in the form of real or continuous numbers. SVR is adequate in overfitting by minimizing the upper bound of the generalization error [7]. Furthermore, the Neural Network (NN) method. NN is a method used to solve discrete, real, and vector problems that resemble the mode of the human nervous system in carrying out its daily tasks [8]. The NN method is often used because it has robust performance in managing various dataset sources. In addition, NN also has the advantage of conducting training based on well-chosen data and can perform calculations in parallel [9]. So that it can be appropriately used to make predictions on data mining.

Several studies have been conducted on the prediction of BRICS member countries' currency exchange rates. Wang et al., (2021) predicted the CNY exchange rate against the dollar by combining Convolutional Neural Networks (CNN) for feature extraction and TLSTM, or Tanh Long Short-Term Memory. As a consequence, CNN-TLSTM outperforms MLP, CNN, RNN, LSTM, and CNN-LSTM in terms of prediction results with MAPE 18.945%, MSE 0.00038, and R^2 of 98.95%. In addition, Hamooni et al., (2022), using the theory of monetary behavior to ascertain the Russian ruble's long-term fair value (1999–2021), obtained the results of the Russian ruble exchange rate over 60 days estimated using the GARCH family model, including GARCH, GJRGARCH, EGARCH, and PGARCH. The IGARCH model was optimized, and the optimal conditional standard deviation was estimated at 0.012 after data collection for 61 days. Soewignjo et al. (2024) also researched the forecast of the Chinese currency exchange rate in relation to the rupiah with the SVR method, resulting in a MAPE of 2.516% for testing data. Acquah et al. (2022) predicted the Ghanaian exchange rate against the USD with the Artificial Neural Network method, giving the results that the NN back propagation model with 7% MAPE and RMSE 0.32274.

This research uses the SVR and NN methods to predict the currency exchange rates of BRICS member countries. This research focuses on China and Russia because of the trade disputes between China and Russia and the United States, which initiated Russia's de-dollarization policy by reducing the use of dollars in the trade balance. This study's innovation is found in the application of these methods in accurately predicting exchange rates. In addition, research related to the prediction of China and Rubel exchange rates is also in line with efforts to support the achievement of SDGs points 10 and 17, which focus on global partnerships in reducing the world's economic disparities, especially for Indonesia, which has just joined the BRICS alliance. It is anticipated that the findings of this investigation would significantly to the currency exchange rate prediction literature. In addition, accurate currency exchange rate predictions can help maintain economic stability and strengthen international trade competitiveness.

II. LITERATURE REVIEW

A. *Exchange Rate*

The exchange rate is a comparison between a country's currency and another country's currency. According to Dornbusch and Fischer, there are four types of exchange rates, namely selling rates, buying rates, middle rates, and flat rates. In this study, the data used is included in the buying rate category, which is the exchange rate that applies when someone buys foreign currency or exchanges foreign currency into rupiah. The buying rate reflects the price banks or financial service providers set when they buy foreign currency from customers [14].

B. *Partial Autocorrelation Function*

The function of partial autocorrelation in time series analysis serves to determine a variable's direct correlation with the previous lag after eliminating the influence of other lags. PACF is used to determine significant lags that can be used as predictor variables in a model [15]. According to [16], the mathematical formulation of the PACF is stated as follows:

$$\gamma_{kk} = Corr(Z_t, Z_{t-k}|Z_{t-1}, \dots, Z_{t-k-1}) \quad (1)$$

$$\alpha_j = \gamma_{k1}\alpha_{j-1} + \gamma_{k2}\alpha_{j-2} + \dots + \gamma_{k(k-1)}\alpha_{j-k-1} + \gamma_{kk}\alpha_{j-k} \quad (2)$$

C. *Uji Terasvirta*

According to research conducted by [17], when it comes to finding nonlinear patterns in data, the Terasvirta test is thought to be the most effective technique. This is supported by its high sensitivity to outliers and very high power test results (close to 1) in identifying cases of nonlinearity. The model with nonlinear characteristics can be expressed in the following way:

$$y_t = \varphi(\boldsymbol{\gamma}'\boldsymbol{w_t}) + \boldsymbol{\beta}'\boldsymbol{w_t} + \varepsilon_t;\ \varepsilon_t \sim IIDN(0, \sigma^2) \quad (3)$$

with,

$\boldsymbol{\beta}'$: $(\beta_0, \dots, \beta_p)$,
$\boldsymbol{w_t}$: $(1, y_{t-1}, \dots, y_{t-p})'$,
$\boldsymbol{\gamma}'$: $(y_0, \boldsymbol{y}')$,
$\boldsymbol{y}'$: $(y_1, \dots, y_p)$,
$\varphi(\boldsymbol{\gamma}'\boldsymbol{w_t})$: $\theta_{0j}\psi(\boldsymbol{\gamma}'\boldsymbol{w_t})$, and
$\psi(\boldsymbol{\gamma}'\boldsymbol{w_t})$: $(1 + \exp(-(\boldsymbol{\gamma}'\boldsymbol{w_t}))^{-1}) - \frac{1}{2}$.

$\boldsymbol{\beta}'$ is the linear weight, while $\boldsymbol{\gamma}'$ is the nonlinear component's weight. In addition, φ is a sigmoid-shaped activation function used in neural networks.

The hidden layer in this model is connected to the activation function σ by the weight θ_{0j}, and the input layer is connected to the hidden layer by the weight $\boldsymbol{\gamma_j}$. If the nonlinear component's value equals zero, then the data can be categorized as linear. Thus, the hypothesis used is as follows:

H_0 : $\theta_{01} = \theta_{02} = \dots = \theta_{0q} = 0$ (data is linear)
H_1 : There is at least one $\theta_{0i} \neq 0$ (data is nonlinear), with $i = 1, 2, \dots, q$, where q is the number of neuron units used.

D. *Support Vector Regression (SVR)*

The SVR method can be used to determine the function f(x) as the best hyperplane as a regression function and minimize the possible error by maximizing the boundary. The concept of the SVR method is to have training data $\{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)\}$ where $x_i \in \mathbb{R}^d$. The goal is to find a function $f(x)$ that has the most deviation ε from the obtained target y_i for all of the training data while also being as flat as practicable. The SVR model's general equation shown below [18]:

$$f(\mathbf{x}) = \boldsymbol{w}^T \varphi(\mathbf{x}) + \mathrm{b} \tag{4}$$

where $\boldsymbol{w}$ is the weight vector, $\varphi(\boldsymbol{x})$ is a function that maps x in the dimensional space, and b represents bias. The regression function $f(x)$ can be appropriately generalized by minimizing the norm of $\boldsymbol{w}$ [18]. This leads to the following solution to the optimization problem:

$$min \frac{1}{2} \, || \, \boldsymbol{w}||^2 \tag{5}$$

with conditions:

$$y_i - \boldsymbol{w}^T \varphi(\boldsymbol{x_i}) - b \;\leq \varepsilon \text{ untuk } i = 1,..,l$$
$$\boldsymbol{w}^T \varphi(\boldsymbol{x_i}) + b - y_i \;\leq \varepsilon \text{ untuk } i = 1,..,l$$

The above optimization is in the case where f exists and approximates all pairs (x_i, y_i) with precision ε. Sometimes, some errors are allowed. By introducing auxiliary variables ζ and ζ^* to get beyond the unenforceable constraints of the optimization problem [19]. Additionally, the equation can be changed into the form shown below:

$$min_w \frac{1}{2} \, ||\boldsymbol{w}||^2 + C \, \sum_{i=1}^{n} \, (\zeta_i + \, \zeta_i{}^*) \tag{6}$$

The constraint function for the optimization problem is as follows:

$$\begin{aligned} &y_i - \boldsymbol{w}^T \varphi(\boldsymbol{x_i}) - b \;\leq \varepsilon + \zeta_i \,, \;\; i = 1,..,l \\ &\boldsymbol{w}^T \varphi(\boldsymbol{x_i}) + b - y_i \leq \varepsilon + \zeta_i{}^*, \;\; i = 1,..,l \\ &\zeta\,, \zeta^* \geq 0 \end{aligned} \tag{7}$$

By solving the optimization problem, it is obtained that:

$$w \; = \sum_{i=1}^{n} (a_i - a_i^*) \varphi(\boldsymbol{x}_i) \tag{8}$$

The SVR function can be defined as follows by using the function φ which is approximated by a kernel function to map the data contained in the vector $\boldsymbol{x}$ from the input space to a higher dimensional feature space:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} (a_i - a_i^*) \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{9}$$

Using the chosen kernel function in SVR shown in Table 1, $K(\boldsymbol{x}_i, \boldsymbol{x})$ must be solved to solve the high-dimensional nonlinear issue.

Table 1 The Kernel Function

Type of Kernel	Mathematical Expression
First degree polynomial	$K(\boldsymbol{x}_i, \boldsymbol{x}) = (\boldsymbol{x}_i{}^T\mathbf{x})$
Polynomial	$K(\boldsymbol{x}_i, \boldsymbol{x}) = (\gamma(\boldsymbol{x}_i{}^T\mathbf{x}) + r)^p$
Radial Basis Funciton (RBF)	$K(\boldsymbol{x}_i, \boldsymbol{x}) = exp(-\gamma\|\boldsymbol{x}_i{}^T\mathbf{x}\|^2)$
Sigmoid	$K(\boldsymbol{x}_i, \boldsymbol{x}) = tanh(-\gamma(\boldsymbol{x}_i{}^T\mathbf{x}) + r)$

This study compares the performance of three kernel functions in SVR, namely Polynomial with degree 2, Sigmoid, and Radial Basis Function (RBF). Three parameters must be determined in order to employ SVR with these kernels namely cost (C), gamma (γ), and epsilon (ε), which are optimized using the grid search method.

E. Neural Network (NN)

An artificial neural network (NN) model has high flexibility and includes various analysis techniques, such as nonlinear dynamic systems made of neurons, discriminant models, nonlinear regression, and data dimension reduction [20]. The network design, learning method, and activation function all affect the NN model. Table 2 lists the many kinds of activation functions that were employed to produce the outcome.

Table 2 Activation Function in NN

Type of Activation	Mathematical Expression
Direct Mapping	$f(x) = x$
Tanh Function	$f(x) = \tanh{(x)}$
Logistic	$f(x) = \dfrac{1}{1 + e^{-x}}$
ReLU	$f(x) = \begin{cases} x, x \geq 0 \\ 0, for\ other\ x \end{cases}$

The extended delta rule is a commonly used NN technique that computes the derivative throughout the backpropagation phase using a chain rule [20]. The backpropagation algorithm consists of two main stages. First, it performs feed-forward, which passes the input values to the network. Second, it calculates the error and modifies the preceding layer's weights. The network structure includes the layers in this stage.

One commonly used architecture in modeling is the Feedforward Neural Network (FFNN). An input layer, one hidden layer, and an interlinked output layer make up this design. The following equation, which was derived from Taylor's study, yields the FFNN architecture's output [21]:

$$\widehat{N}_t = f^0 \left\{ v_0 + \sum_{k=1}^{K} \left(v_k f^h \left(\sum_{i=1}^{I} w_{ik} x_{i,t} \right) \right) \right\} \tag{10}$$

In this case, v_k represents the weight of the neuron to$-k$ in the hidden layer, while w_{ik} is the weight connecting the i-th input to the k-th neuron in the hidden layer. The activation functions used in the hidden layer and output layer are denoted as $f^h()$ and $f^0()$ respectively. Besides Feedforward Neural Network (FFNN), there is another architecture called Deep Learning Neural Network (DLNN). Unlike the FFNN, the DLNN architecture uses two hidden layers to connect the layer that receives input and the output layer, thus allowing the capture of more complex patterns in the data. One form of DLNN that is often used in NN modeling is the Perceptron with Multi Layer (MLP). The structure of the neural network model with Multi-Layer Perceptron (MLP) is shown as follows in Figure 1:

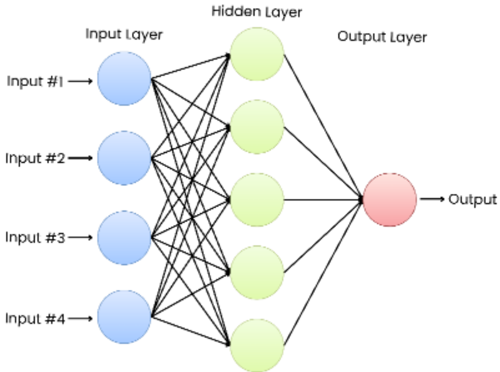


Figure 1 Neural Network Structure

Within the neural network architecture shown in Figure 1, the hidden layer serves as the feature extraction stage. This layer is where the activation function is applied, and the weights are weighted. Suppose $\boldsymbol{V}_{ik}$ is the weight matrix that joins the hidden layer to the input, while $\boldsymbol{d}_i = (d_1, d_2, \ldots, d_i)'$ represents a number of lags with a particular significance. In addition, $\boldsymbol{b}_k = (b_1, b_2, \ldots, b_k)'$ is a bias vector that adjusts the input received. Each input given weight will be calculated by summing all signals that enter the hidden layer using Equation (11) as follows:

$$\boldsymbol{z}_{in_k} = \sum_{i=1}^{n} \boldsymbol{d}_i \,.\, \boldsymbol{V}_{ik} + \boldsymbol{b}_k \tag{11}$$

After this summation, the activation function is applied to each unit in the hidden layer, thus obtained as presented in Equation (12) as follows:

$$\boldsymbol{z}_k = g(\boldsymbol{z}_{in_k}) \tag{12}$$

In the hidden layer units, g is the activation function. The output at the output layer is then predicted using the findings from the hidden layer. New weights are assigned at this stage, and the activation function is again calculated as follows:

$$\boldsymbol{y}_{in_m} = \sum_{k=1}^{p} \boldsymbol{z}_k \,.\, \boldsymbol{V}_{km} \tag{13}$$

$$\boldsymbol{y}_m = g(\boldsymbol{y}_{in_m}) \tag{14}$$

with,

$\boldsymbol{V}_{km}$: weight matrix between the m-th output and the k-th hidden layer,
$\boldsymbol{z}_k$: the buried layer's output vector,
$\boldsymbol{y}_{in_m}$: the output value vector prior to the application of the activation function, and
$\boldsymbol{y}_m$: the final output result vector following the application of activation.

After the $\boldsymbol{y}_m$ value is obtained as a prediction result, this value and the real value are contrasted, thus obtaining the error value presented in Equation (15) as follows:

$$\boldsymbol{\varepsilon}_m = \boldsymbol{y}_m - \boldsymbol{t}_m \tag{15}$$

With $\boldsymbol{t}_m$ as the actual value. Then, a gradient calculation is performed on $\boldsymbol{y}_m$, which is expressed in Equation (16) as follows:

$$\boldsymbol{\delta}_m = (\boldsymbol{y}_m - \boldsymbol{t}_m) \,.\, g'(\boldsymbol{y}_{in_m}) \tag{16}$$

With $g'(\boldsymbol{y}_{in_m})$ as the activation function's derivative in the output layer. To update the weights, use the learning rate η with the formula stated in Equation (17) as follows:

$$\boldsymbol{V}_{km}^{new} = \boldsymbol{V}_{km} + \eta \,.\, \boldsymbol{z}_k \,.\, \boldsymbol{\delta}_m \tag{17}$$

for the hidden layer, the error gradient is calculated with Equation (18) as follows:

$$\boldsymbol{\delta}_k = \sum_{m=1}^{q} \boldsymbol{V}_{km} \,.\, \boldsymbol{\delta}_m \,.\, g'(\boldsymbol{z}_{in_k}) \tag{18}$$

while the weight update on the hidden layer is done with Equation (19) as follows:

$$\boldsymbol{V}_{ik}^{new} = \boldsymbol{V}_{ik} + \eta \,.\, \boldsymbol{d}_i \,.\, \boldsymbol{\delta}_k \tag{19}$$

This stage is repeated until the error value becomes minimal (convergent) or a specific iteration limit is reached.

F. Walking Forward Validation

In determining the optimal parameters in time series analysis, an algorithm that can properly separate the data into training and testing is required. Usually, the search for optimal parameters is done using the grid search method, which is commonly paired with K-fold cross-validation. However, in time series data, using K-fold cross-validation is unsuitable because it causes the testing data to be out of sequence [22]. In the case of time series, the testing data should always come after the training data to maintain temporal order [23]. Therefore, a more suitable validation method is forward-walking validation [24]. The tuning parameter values are presented in Table 3 below.

Table 3 Parameter Tuning Value of Each Algorithm

Algorithm	Tuning Parameters	Reference
SVR	Kernel: [RBF, Sigmoid, Polynomial], C: [0.1; 1; 10], Epsilon: [0.01; 0.1; 0,2], Gamma: [0.1; 1; 10],	[25]
Neural Network	Hidden layer size: [(50,), (100,), (150,), (200,), (250,), (300,), (50, 50), (100,100), (150,150), (200,200), (250,250), (300,300)], Activation function: [Tanh, Logistic, ReLU], Learning rate: [0.0001, 0.001, 0.01]	[26] [27]

G. Data Normalization

Normalization aims to adjust data to fit certain functions with a limited output range, such as sigmoid and tanh. This process generates new values within a specific range to ensure the data stays within the desired limits. In this case, the minimum and maximum values of the data can be found in Table 6, while the new range used is 0.1 to 1 to avoid zero values. The equation used for normalization is as follows [28]:

$$y_t' = \frac{y_t - min(data)}{max(data) - min(data)}\left(new_{max(data)} - new_{min(data)}\right) + new_{min(data)} \tag{20}$$

with,

y_t' : data after normalized at time t, and

y_t : data before normalized at time t.

H. Kriteria Model Terbaik

1. Mean Squared Error (MSE)

MSE is a metric used to calculate the average of the squared difference between predicted and actual values [29]. The MSE formula can be expressed as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - F(X_i)\right)^2 \tag{21}$$

with,

N : total number of observations,

Y_i : actual value,

$F(X_i)$: predicted value of the model for the *i*-th observation, and

$\left(Y_i - F(X_i)\right)^2$: the square of the difference between the actual value and the predicted value.

The MSE value shows how accurate the model is in making predictions. The smaller the MSE value, the better the model performance in predicting the actual value.

2. Mean Absolute Error (MAE)

This metric calculates the average absolute value of the difference between the predicted and actual values, regardless of the direction of the error (either positive or negative) [30]. The MAE formula is expressed as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - F(X_i)| \tag{22}$$

with,

n : total number of observations,

Y_i : actual value,

$F(X_i)$: predicted value of the model for the *i*-th observation, and

$Y_i - F(X_i)$: the absolute difference between the actual value and the predicted value.

MAE provides an overview of the average prediction error in the same unit as the original data. The smaller the MAE value, the better the model's performance in making predictions.

3. Mean Absolute Percentage Error (MAPE)

MAPE measures the average absolute percentage error between predicted and actual values. It evaluates prediction accuracy by calculating the absolute difference, expressed as a percentage [31]. The equation for calculating MAPE is presented in Equation (23) as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \tag{23}$$

with,

y_i : exchange rate price at the time *i*,

$\hat{y}_i$: prediction of the exchange rate price at the time *i*, and

n : number of predicted data.

The following is the category of the prediction accuracy level of the MAPE value:

Table 4 MAPE Value Categories

MAPE	Prediction Accuracy Level
$MAPE < 10\%$	Very good
$10\% \leq MAPE < 20\%$	Good
$20\% \leq MAPE < 50\%$	Simply
$MAPE \geq 50\%$	Bad

III. METHODOLOGY

A. Data and Data Sources

This research uses quantitative research emphasizing theory testing through measuring research variables with numbers and analyzing data with predetermined statistical procedures. This research uses secondary data in the form of USD against CNY and USD against RUB exchange rates obtained from the *Investing.com* page as weekly data.

The data is divided into two parts, namely training data and testing data, to build and evaluate the developed prediction model. The data is divided into 90% training and 10% testing data. So 90% of the total 812 data is used as training data, which is 730 data covering the period from June 21, 2009, to June 11, 2023. As for the testing data, 82 of the 812 data were used, covering the period from June 18, 2023, to February 9, 2025. Thus, this study uses two types of exchange rate data, namely the USD rate against CNY and the USD rate against RUB. The variables in this study are presented in Table 5 as follows:

Table 5 Research Variables

Variables	Description	Unit
Y_1	USD-RUB Rate	RUB (Weekly)
Y_2	USD-CNY Rate	CNY (Weekly)

B. Data Analysis Stages

The stages that will be carried out in data analysis are as follows:

1. The descriptive statistical analysis stage of the course to find out general information about the data.
   a. Present data visualization in the form of a time series plot using a line chart.
   b. Analyze data characteristics and display descriptive statistics calculations.
2. The stages of modelling the exchange rate using the Support Vector Regression (SVR) method are as follows:
   a. Determine the influential lag by analyzing the PACF plot according to Equation (1) and the adjusting the shape of the data afterwards by normalizing the data based on Equation (20).
   b. Test the linearity of the relationship between variables using the Terasvirta test based on Equation (4) and identify significant lags to determine the most appropriate kernel type for the SVR model.
   c. Determine the optimal parameters in the SVR model, namely the value of gamma (γ), cost (C), and epsilon (ε) select the most suitable kernel type using the walking forward validation approach.
   d. Using the SVR model that has been built to predict the USD-RUB rate and USD-CNY rate based on testing data.
   e. Evaluate the performance of the prediction model using several evaluation metrics, namely MSE based on Equation (21), MAE based on Equation (22), and MAPE based on Equation (23).
3. The stage of modelling the exchange rate using the Neural Network (NN) method are as follows:
   a. Determine the influential lag by analyzing the PACF plot according to Equation (1) and then adjusting the shape of the data afterwards by normalizing the data based on Equation (20).
   b. Test the linearity of the relationship between variables using the Terasvirta test based on Equation (4) and identify significant lags to determine the most appropriate activation function for the NN model.
   f. Determine the optimal parameters for the NN architecture, including the hidden layer size value, the type of activation function used, and the learning rate value using walking forward validation.
   c. Using the NN model that has been built to predict the USD-RUB rate and USD-CNY rate based on testing data.
   d. Evaluate the performance of the prediction model using several evaluation metrics, namely MSE based on Equation (21), MAE based on Equation (22), and MAPE based on Equation (23).
4. Comparing the performance of SVR and NN models in predicting USD-RUB rate and USD-CNY rate based on MSE, MAE, and MAPE values to determine the model with the best accuracy.

IV. RESULTS AND DISCUSSIONS

A. Descriptive Statistics

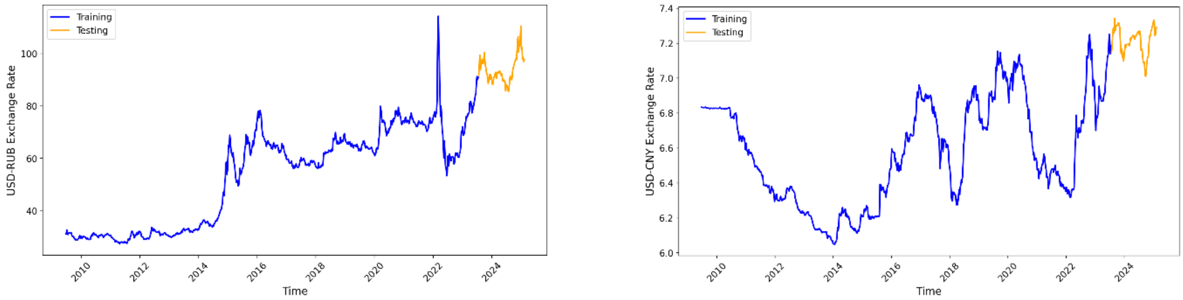
Based on the USD-RUB rate and USD-CNY rate data obtained from the *Investing.com* page, Table 6 presents an overview of the 812 weekly scale data from June 21, 2009, to February 9, 2025.

Table 6 Descriptive Statistics

Variables	Mean	Standard Deviation	Minimum	Maximum
USD-RUB Rate	57.339	21.376	27.407	114.253
USD-CNY Rate	6.6394	0.3434	6.0488	7.3430

Based on Table 6, the minimum value of the USD-RUB rate was recorded at 27.407 on April 24, 2011. During this period, the Russian economy was still stable with high oil prices, which supported strengthening the ruble against the US dollar. According to a Directorate General of Oil and Gas report, the Indonesian Crude oil Price (ICP) in March 2011 reached USD 113.07 per barrel. This increase was in line with the rise in world oil consumption, which increased by 1.5 million barrels compared to the previous year. As a major oil producer, Russia benefited from high oil prices, which contributed to strengthening the ruble against the US dollar. In contrast, the maximum value of the USD-RUB rate reached 114.253 on March 6, 2022, which occurred after Russia invaded Ukraine in February 2022. The economic sanctions imposed by Western countries against Russia led to a drastic weakening of the ruble, so the USD-RUB rate jumped sharply.

Meanwhile, the minimum value of the USD-CNY rate of 6.0488 occurred on January 19, 2014. In this period, the Chinese economy was experiencing strong growth with a high trade surplus, so the yuan currency strengthened against the US dollar. In contrast, the maximum value of the USD-CNY rate reached 7.3430 on September 3, 2023. This increase occurred amid global economic uncertainty and pressure on the yuan due to China's economic slowdown and monetary policy measures taken by the US Federal Reserve, which led to the strengthening of the dollar against global currencies, including the yuan. The time series plots of the USD-RUB rate and the USD-CNY rate, with the division of the training period at 90% and testing at 10%, are presented in Figure 2.



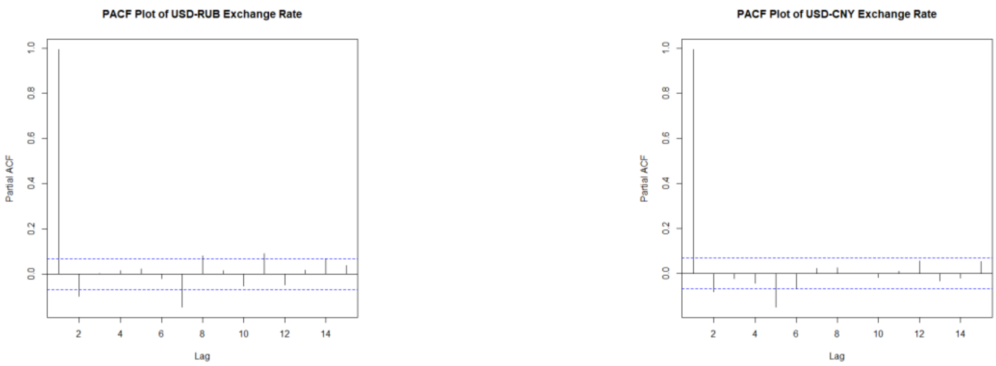
(a) (b)

Figure 2 Time Series Visualization of USD-RUB Rate and USD-CNY Rate

In Figure 2(a), the USD-RUB rate shows a volatile trend with several significant spikes. In the period around 2014 and 2015, there was a sharp spike that coincided with the fall in world oil prices. Furthermore, in the period 2021 to early 2022, the USD-RUB rate tends to show a downward trend. However, from mid-2022 until the end of 2023, the USD-RUB rate experienced a significant upward trend. The USD-RUB rate does not show a clear trend because it fluctuates up and down, so the data is non-stationary and does not have a seasonal pattern (non-seasonal). Meanwhile, in Figure 2(b), the USD-CNY rate shows a relatively more stable trend than the USD-RUB rate. It experienced a downward trend from 2010 to 2014 before showing an upward trend in the following years. After 2018, the USD-CNY rate tends to experience a continuous increase. The USD-CNY rate showed high volatility in the testing period around 2023 to 2024. Overall, the USD-CNY rate data is also non-stationary, with a non-seasonal pattern and fluctuations influenced by global economic dynamics

B. Determination of Input Lag

Determining the input lag in time series analysis is important to identify significant lags as predictor variables. This process is done by analyzing the Partial Autocorrelation Function (PACF) plot, which measures the partial correlation between variables in the time series. The results of the PACF plot showing significant lags are presented in Figure 3.



(a) (b)

Figure 3 PACF Plot of USD-RUB Rate and USD-CNY Rate

Figure 3(a) presents the PACF plot for the USD-RUB rate, which shows that the 1st, 2nd, 7th, 8th, and 11th lag significantly influences the USD-RUB rate data. Therefore, in modeling the USD-RUB rate, the sharing of y_t data will begin in the $t = 11$ periods to ensure that all significant lags can be utilized as predictor variables in the model. Figure 3(b) presents the PACF plot for the USD-CNY rate, which shows that the 1st, 2nd, and 5th lag significantly influence the USD-CNY rate data. Therefore, in modeling the USD-CNY rate, the sharing of y_t data will begin in the $t = 5$ periods to ensure that all significant lags can be utilized as predictor variables in the model.

C. Nonlinearity Testing

This research uses quantitative research, emphasizing theory testing by measuring research variables with numbers and analyzing data with predetermined statistical procedures. It also uses secondary data in the form of USD against RUB and USD against CNY exchange rates obtained from the Investing.com page in the form of weekly data.

Table 7 Terasvirta Test Results

Variables	Lag	Test Statistics	P-Value
USD-RUB Rate	1	$F_{(0.05;2;814)} = 6.984$	0.001
USD-CNY Rate	1	$F_{(0.05;2;814)} = 0.7035$	0.4951
	2	$F_{(0.05;7;808)} = 3.5149$	0.001

Based on Table 7, the Terasvirta test results show that for the USD-RUB rate at lag 1 and the USD-CNY rate at lag 2, has p-value is not greater than the significant level $\alpha = 5\%$, so H_0 is rejected and indicates that the data has a nonlinear pattern. Thus, kernel functions suitable for handling nonlinear patterns in SVR modeling are the Radial Basis Function (RBF), polynomial, and sigmoid. Meanwhile, in neural network modeling, activation functions that can be used include logistic, reLU, and tanh to handle the complexity of nonlinear data.

D. Support Vector Regression (SVR) Prediction

1. USD-RUB Rate

The results of the Terasvirta test, which indicated the existence of non-linear data patterns in the USD-RUB rate data, became the basis for selecting nonlinear kernel functions, namely Radial Basis Function (RBF), polynomial, and sigmoid. Before applying SVR modeling with the grid search method, walking forward validation, and using the parameter values determined in Table 3 to get the best parameter combination, the data was first normalized according to Equation (20). The optimal parameter results using SVR modeling for the USD-RUB rate are presented in Table 8.

Table 8 Optimal SVR Parameters USD-RUB Rate

Kernel Type	Parameters	Optimal Value
RBF	Cost (C)	10
	Gamma (γ)	0.1
	Epsilon (ε)	0.01
Sigmoid	Cost (C)	1
	Gamma (γ)	0.1
	Epsilon (ε)	0.01
Polynomial	Cost (C)	0.1
	Gamma (γ)	1
	Epsilon (ε)	0.01

After the optimal parameters are determined, the following step involves evaluating the model using testing data that was not included during the training phase. This stage assesses the model's ability to generalize new data that has never been learned. After the prediction is made, the results are returned to the initial scale through the denormalization process so that they can be compared directly with the actual value. The results of the prediction model performance evaluation for the USD-RUB rate using various types of kernels are presented in Table 9.

Table 9 SVR Prediction Results USD-RUB Rate

Kernel Type	Model Criteria	Evaluation Results
RBF	MSE	13.9723
	MAE	3.1089
	MAPE	3.24%
Sigmoid	MSE	6.1596
	MAE	1.8808
	MAPE	1.95%
Polynomial	MSE	555.3671
	MAE	22.6478
	MAPE	23.96%

Table 9 presents the prediction results using the SVR method for the USD-RUB rate, which shows that the model with the sigmoid kernel shows the best performance compared to other kernels with optimal parameters, namely Cost (C) of 1, *Gamma* (γ) of 0.1, and *Epsilon* (ε) of 0.01. This is indicated by the MSE value of 6.1596, MAE of 1.8808, and MAPE of 1.95%, which shows a lower error rate than other kernels.

2. USD-CNY Rate

The results of the Terasvirta test, which indicated the existence of non-linear data patterns in the USD-CNY rate data, became the basis for selecting nonlinear kernel functions, namely Radial Basis Function (RBF), polynomial, and sigmoid.

Before applying SVR modeling with the grid search method, walking forward validation, and using the parameter values determined in Table 3 to get the best parameter combination, the data was first normalized according to Equation (20). The optimal parameter results using SVR modeling for the USD-CNY rate are presented in Table 10.

Table 10 Optimal SVR Parameters USD-CNY Rate

Kernel Type	Parameters	Optimal Value
RBF	Cost (C)	10
	Gamma (γ)	0.1
	Epsilon (ε)	0.01
Sigmoid	Cost (C)	1
	Gamma (γ)	0.1
	Epsilon (ε)	0.01
Polynomial	Cost (C)	1
	Gamma (γ)	1
	Epsilon (ε)	0.01

The results of the prediction model performance evaluation for the USD-CNY rate using various types of kernels are presented in Table 11.

Table 11 SVR Prediction Results USD-CNY Rate

Kernel Type	Model Criteria	Evaluation Results
RBF	MSE	0.0014
	MAE	0.0322
	MAPE	0.45%
Sigmoid	MSE	0.0015
	MAE	0.0329
	MAPE	0.46%
Polynomial	MSE	0.0517
	MAE	0.2153
	MAPE	2.98%

Table 11 presents the prediction results using the SVR method for the USD-CNY rate, which shows that the model with the Radial Basis Function (RBF) kernel shows the best performance compared to other kernels with optimal parameters, namely Cost (C) of 10, *Gamma* (γ) of 0.1, and *Epsilon* (ε) of 0.01. This is indicated by the MSE value of 0.0014, MAE of 0.0322, and MAPE of 0.45%, which shows a lower error rate than other kernels.

E. Neural Network Prediction

1. USD-RUB Rate

The selection of the neural network model is based on the nonlinear data test results obtained through the Terasvirta test, which indicates that the activation function used must also be nonlinear, such as logistic, reLU, and tanh. Before conducting neural network modeling with the grid search approach, walking forward validation, and using the parameter values listed in Table 3, the data is first normalized according to Equation (20). The optimal parameter results using neural network modeling for the USD-RUB rate are presented in Table 12.

Table 12 Optimal Neural Network Parameters USD-RUB Rate

Activation Function	Parameters	Optimal Value
Tanh	Learning rate	0.01
	Hidden layer size	250;250
Logistic	Learning rate	0.0001
	Hidden layer size	300;
ReLU	Learning rate	0.01
	Hidden layer size	250;250

The results of the prediction model performance evaluation for the USD-RUB rate using various activation function are presented in Table 13.

Table 13 Neural Network Prediction Results USD-RUB Rate

Activation Function	Model Criteria	Evaluation Results
Tanh	MSE	7.2694
	MAE	2.0629
	MAPE	2.14%
Logistic	MSE	16.8756
	MAE	3.1979
	MAPE	3.33%

Activation Function	Model Criteria	Evaluation Results
ReLU	MSE	8.0217
	MAE	2.1822
	MAPE	2.27%

Table 13 presents the prediction results using the neural network method for the USD-RUB rate, which shows that the model with the tanh activation function shows the best performance compared to other activation functions with its optimal parameters, namely a learning rate of 0.01 with a hidden layer size of 250;250. This is shown by the MSE value of 7.2694, MAE of 2.0629, and MAPE of 2.14%, which shows a lower error rate than other activation functions. Furthermore, the illustration of the neural network model with the optimal parameter value of hidden layer sizes 250 and 250 is presented in Figure 4 as follows:

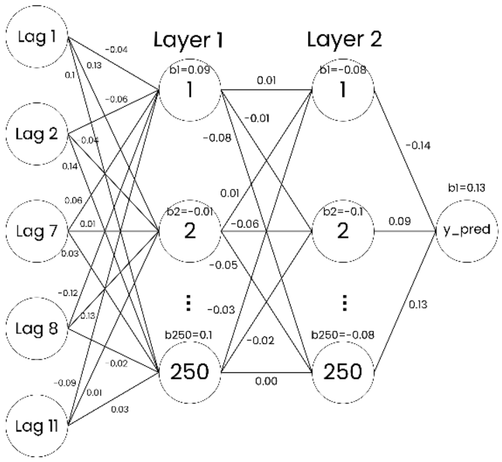


Figure 4 Neural Network Structure of USD-RUB Rate

In the designed neural network model, each input d_i from lag 1, 2, 7, 8, and 11 data will be connected to each neuron in hidden layer 1 through predetermined weights. Activation in hidden layer 1 is calculated by summing up the multiplication result between each input and its weight, then adding it with the bias value. The result will be passed through the tanh activation function to produce the output value in the first layer. The calculation for the k-th neuron in hidden layer 1 is mathematically expressed in Equation (24) as follows:

$$\boldsymbol{g}_k^{(1)} = \phi\left(\sum_{i\in\{1,2,7,8,11\}} \boldsymbol{V}_{ik}^{(1)}.\boldsymbol{d}_i + \boldsymbol{b}_k^{(1)}\right), k = 1,2,\dots,250 \quad (24)$$

with,

ϕ : activation function,
d_i : input lags 1, 2, 7, 8 and 11,
v_{ik} : the weight of the input d_i to the k-th, and
b_k : bias at the k-th node.

For example, the calculation for a node in hidden layer 1 can be expressed based on Equation (24) as follows:

$$g_1^{(1)} = Tanh\left(v_{1,1}^{(1)}.d_1 + v_{2,1}^{(1)}.d_2 + v_{7,1}^{(1)}.d_7 + v_{8,1}^{(1)}.d_8 + v_{11,1}^{(1)}.d_{11} + b_1^{(1)}\right)$$

After obtaining the activation value of hidden layer 1, each neuron in the layer will be connected to each neuron in hidden layer 2. The calculation in this layer also follows the same pattern, which is to add up the results of multiplying the weights with the activation value of the previous layer, add a bias, then apply the tanh activation function. The Equation can be written in Equation (25) as follows:

$$\boldsymbol{g}_m^{(2)} = \phi\left(\sum_{k=1}^{250} \boldsymbol{g}_k^{(1)}.\boldsymbol{V}_{km}^{(2)} + \boldsymbol{b}_m^{(2)}\right), m = 1,2,\dots,250 \quad (25)$$

For example, the calculation for one of the neurons in hidden layer 2 based on Equation (25) can be expressed as follows:

$$g_1^{(2)} = Tanh\left(v_{1,1}^{(2)}.g_1^{(1)} + v_{2,1}^{(2)}.g_2^{(1)} + \dots + v_{250,1}^{(2)}.g_{250}^{(1)} + b_1^{(2)}\right)$$

Furthermore, the results of hidden layer 2 will be used as input to the output layer to produce the final prediction value ($\hat{y}$). The calculation in this layer is done in the same way, namely summing up the results of multiplying the weights with activation from the previous layer, plus bias, then passed to the activation function to produce the final output value. The equation can be written in Equation (26) as follows:

$$\hat{\boldsymbol{y}} = \phi\left(\sum_{m=1}^{250} \boldsymbol{v}_m^{(output)}.\boldsymbol{g}_m^{(2)} + b^{(output)}\right) \quad (26)$$

For example, the calculation for the predicted value based on Equation (26) can be expressed as follows

$$\hat{\boldsymbol{y}} = Tanh\left(v_1^{(output)}.g_1^{(2)} + v_2^{(output)}.g_2^{(2)} + \dots + v_{250}^{(output)}g_{250}^{(2)} + b^{(output)}\right)$$

After the feedforward stage is performed, the next stage is backpropagation, which aims to adjust the weights using the gradient descent method. In each layer, the error value is calculated using the formula in Equation (27) as follows:

$$\boldsymbol{\delta}^{(output)} = (\hat{\boldsymbol{y}} - \boldsymbol{y}_{actual}) \cdot \phi' \quad (27)$$

with,

$\hat{y}$: prediction result,
y_{actual} : the actual target value, and
ϕ' : the first derivative of the activation function.

Furthermore, the error calculated at the output layer will be forwarded to hidden layer 2 using Equation (28) as follows:

$$\boldsymbol{\delta}_m^{(2)} = \boldsymbol{\delta}^{(output)} \,.\, \boldsymbol{v}_m^{(output)} \,.\, \boldsymbol{\phi}' \quad (28)$$

where $m = 1, 2, \dots, 250$ represents the number of neurons in hidden layer 2. Then, this error is also calculated for hidden layer 1 by summing the contributions from all neurons in hidden layer 2. The equation is presented in Equation (29) as follows:

$$\boldsymbol{\delta}_k^{(1)} = \sum_{m=1}^{250} \boldsymbol{V}_{km}^{(2)} . \boldsymbol{\delta}_m^{(2)} \,.\, \boldsymbol{\phi}' \quad (29)$$

where $k = 1, 2, \dots, 250$ is the neuron index in hidden layer 1. After obtaining the error value (δ), the weights will be updated using Equation (30) and Equation (31) as follows:

$$\boldsymbol{V}_{(new)}^{(layer)} = \boldsymbol{V}_{(old)}^{(layer)} - \eta \,.\, \boldsymbol{\delta}^{(layer)} . \boldsymbol{z} \quad (30)$$

$$\boldsymbol{b}_{(new)}^{(layer)} = \boldsymbol{b}_{(old)}^{(layer)} - \eta \,.\, \boldsymbol{\delta}^{(layer)} \quad (31)$$

with,
η : learning rate, and
$\boldsymbol{z}$: input to each layer.

The process will continue to repeat until the weights reach the optimal value or the maximum number of iterations so that the model can produce predictions with higher accuracy.

2. *USD-CNY Rate*

The selection of the neural network model is based on the nonlinear data test results obtained through the Terasvirta test, which indicates that the activation function used must also be nonlinear, such as logistic, reLU, and tanh. Before conducting neural network modeling with the grid search approach, walking forward validation, and using the parameter values listed in Table 3, the data is first normalized according to Equation (20). The optimal parameter results using neural network modeling for the USD-CNY rate are presented in Table 14.

Table 14 Optimal Neural Network Parameters USD-CNY Rate

Activation Function	Parameters	Optimal Value
Tanh	Learning rate	0.01
	Hidden layer size	100;100
Logistic	Learning rate	0.0001
	Hidden layer size	300;300
ReLU	Learning rate	0.001
	Hidden layer size	300;300

The results of the prediction model performance evaluation for the USD-CNY rate using various activation function are presented in Table 15.

Table 15 Neural Network Prediction Results USD-CNY Rate

Activation Function	Model Criteria	Evaluation Results
Tanh	MSE	0.0019
	MAE	0.0366
	MAPE	0.51%
Logistic	MSE	0.0021
	MAE	0.0344
	MAPE	0.48%
ReLU	MSE	0.0017
	MAE	0.0345
	MAPE	0.48%

Table 15 presents the prediction results using the neural network method for the USD-CNY rate, which shows that the model with the reLU activation function shows the best performance compared to other activation functions with its optimal parameters, namely a learning rate of 0.001 with a hidden layer size of 300;300. This is shown by the MSE value of 0.0017, MAE of 0.0345, and MAPE of 0.48%, which shows a lower error rate than other activation functions. Furthermore, the illustration of the neural network model with the optimal parameter value of hidden layer sizes 300 and 300 is presented in Figure 5 as follows:

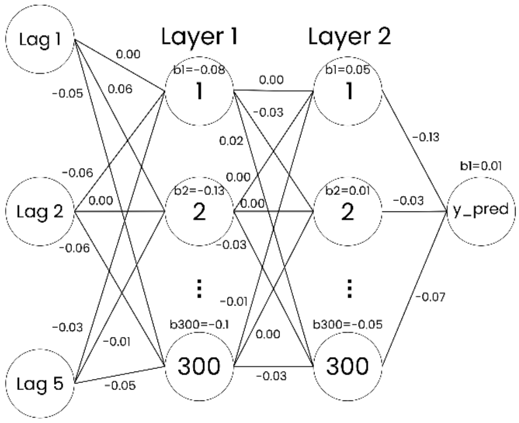


Figure 5 Neural Network Structure of USD-CNY Rate

In the designed neural network model, each input d_i from lag 1, 2, and 5 data will be connected to each neuron in hidden layer 1 through predetermined weights. Activation in hidden layer 1 is calculated by summing up the multiplication result between each input and its weight, then adding it with the bias value. The result will be passed through the reLU activation function to produce the output value in the first layer. The calculation for the k-th neuron in hidden layer 1 is mathematically expressed in Equation (32) as follows:

$$\boldsymbol{g}_k^{(1)} = \phi\left(\sum_{i\in\{1,2,5\}} \boldsymbol{V}_{ik}^{(1)}.\boldsymbol{d}_i + \boldsymbol{b}_k^{(1)}\right), k = 1,2,\dots,300 \tag{32}$$

with,

ϕ : activation function,
d_i : input lags 1, 2, and 5,
v_{ik} : the weight of the input d_i to the k-th node, and
b_k : bias at the k-th node.

For example, the calculation for a node in hidden layer 1 can be expressed based on Equation (32) as follows:

$$g_1^{(1)} = ReLU\left(v_{1,1}^{(1)}.d_1 + v_{2,1}^{(1)}.d_2 + v_{5,1}^{(1)}.d_5 + b_1^{(1)}\right)$$

After obtaining the activation value of hidden layer 1, each neuron in the layer will be connected to each neuron in hidden layer 2. The calculation in this layer also follows the same pattern, namely summing the results of multiplying the weights with the activation value of the previous layer, adding a bias, then applying the reLU activation function. The equation can be written in Equation (33) as follows:

$$\boldsymbol{g}_m^{(2)} = \phi\left(\sum_{k=1}^{300} \boldsymbol{g}_k^{(1)}.\boldsymbol{V}_{km}^{(2)} + \boldsymbol{b}_m^{(2)}\right), m = 1,2,\dots,300 \tag{33}$$

For example, the calculation for one of the neurons in hidden layer 2 based on Equation (38) can be expressed as follows:

$$g_1^{(2)} = ReLU\left(v_{1,1}^{(2)}.g_1^{(1)} + v_{2,1}^{(2)}.g_2^{(1)} + \dots + v_{300,1}^{(2)}.g_{300}^{(1)} + b_1^{(2)}\right)$$

Furthermore, the results of hidden layer 2 will be used as input to the output layer to produce the final prediction value ($\hat{y}$). The calculation in this layer is done in the same way, namely summing up the results of multiplying the weights with activation from the previous layer, plus bias, then passed to the activation function to produce the final output value. The equation can be written in Equation (34) as follows:

$$\hat{\boldsymbol{y}} = \phi\left(\sum_{m=1}^{300} \boldsymbol{v}_m^{(output)}.\boldsymbol{g}_m^{(2)} + b^{(output)}\right) \tag{34}$$

For example, the calculation for the predicted value based on Equation (34) can be expressed as follows:

$$\hat{\boldsymbol{y}} = ReLU\left(v_1^{(output)}.g_1^{(2)} + v_2^{(output)}.g_2^{(2)} + \dots + v_{300}^{(output)}g_{300}^{(2)} + b^{(output)}\right)$$

After the feedforward stage, backpropagation is performed to adjust the weights using the gradient descent method. The error calculation at each layer follows the same process as the USD-RUB rate, with a two-layer model already in use.

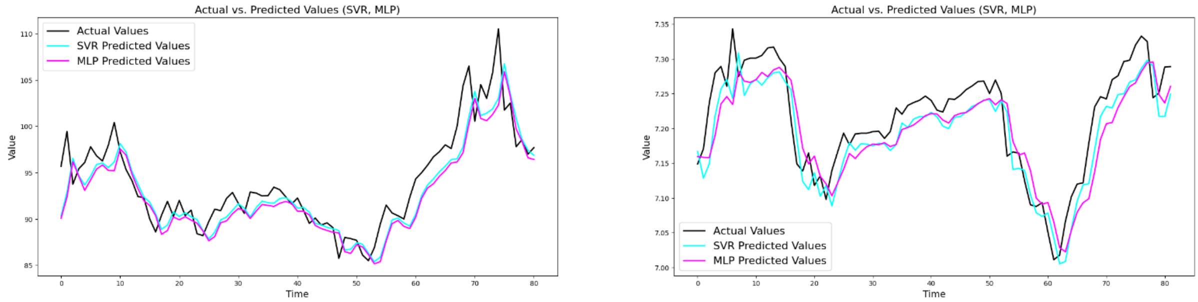
F. Best Model Comparison

After the modeling process is carried out on all models, the next step is to compare the prediction results of each model to determine the model with the best performance. The results of the performance comparison of each model on the USD-RUB rate and USD-CNY rate are presented in Table 16 as follows:

Table 16 Comparison of Model Performance in Exchange Rates Prediction

Variables	Model	Evaluation Results
USD-RUB Rate	SVR with kernel sigmoid	MSE: 6.1596 ; MAE: 1.8808 ; MAPE: 1.95%
	Neural Network with tanh activation function	MSE: 7.2694 ; MAE: 2.0629 ; MAPE: 2.14%
USD-CNY Rate	SVR with kernel RBF	MSE: 0.0014 ; MAE: 0.0322 ; MAPE: 0.45%
	Neural Network with reLU activation function	MSE: 0.0017 ; MAE: 0.0345 ; MAPE: 0.48%

Based on the evaluation results presented in Table 16, it can be concluded that the SVR method with sigmoid kernel is superior in predicting the USD-RUB rate, while SVR with RBF kernel provides the best prediction results for the USD-CNY rate. The comparison plot of USD-RUB rate and USD-CNY rate predictions is presented in Figure 6 as follows:



(a) (b)

Figure 6 Comparison Plot of Predicted USD-RUB Rate and USD-CNY Rate

Based on Figure 6(a) which displays a comparison between the actual value and the prediction results of the Support Vector Regression (SVR) and neural network methods for the USD-RUB rate, it can be observed that the SVR model tends to be closer to the actual value than the neural network model. This can be seen from the blue line pattern (SVR) which more often follows the fluctuation of the black line (actual value) compared to the pink line (neural network). Meanwhile, Figure 6(b) shows the prediction results of the USD-CNY rate using the SVR and neural network methods. In this graph, both models show excellent performance in following the movement pattern of the actual value, with the SVR slightly superior at some points. Based on these two graphs, it can be concluded that the SVR method has an advantage in predicting the movement of the USD-RUB rate and the USD-CNY rate compared to the neural network.

V. CONCLUSIONS AND SUGGESTIONS

Based on the analysis, the USD-RUB and USD-CNY rates show different movement characteristics from 2009 to 2025. The USD-RUB rate experienced higher fluctuations with significant volatility due to various external factors, such as oil prices and economic sanctions. Meanwhile, the USD-CNY rate shows a more stable movement, although it is still influenced by monetary policy and global economic conditions. The results of this study show that the Support Vector Regression (SVR) method is the best approach in predicting the USD-RUB rate and the USD CNY rate compared to the Neural Network. For the USD-RUB rate, the SVR model with sigmoid kernel provides prediction performance with MSE value of 6.1596, MAE of 1.8808, and MAPE of 1.95%, which indicates that this model can capture the pattern of exchange rate movement very well. Meanwhile, for the USD-CNY rate, the SVR model with Radial Basis Function (RBF) kernel produces more accurate predictions with MSE value of 0.0014, MAE of 0.0322, and MAPE of 0.45%, indicating that this model is effective in capturing the movement pattern of the USD-CNY rate which is more stable than USD-RUB rate. Future research is recommended to develop a hybrid method that combines SVR and Neural Network to improve prediction accuracy. In addition, macroeconomic variables such as inflation, interest rates, and trade policies must be considered to understand the factors influencing exchange rate movements. This development is expected to produce a more accurate prediction model to support decision making in the economic and financial fields.

VI. REFERENCES

[1] M. Saaida, “BRICS Plus: de-dollarization and global power shifts in new economic landscape,” *BRICS Journal of Economics*, vol. 5, no. 1, pp. 13–33, Apr. 2024, doi: 10.3897/brics-econ.5.e117828.

[2] T. D. Arnold, “De-dollarization and global sovereignty: BRICS’ quest for a new financial paradigm,” *Human Geography*, Aug. 2024, doi: 10.1177/19427786241266896.

[3] E. Cerutti, G. Gopinath, and A. Mohommad, “The Impact of US-China Trade Tensions,” IMF Blog.

[4] M. Amiti, S. J. Redding, and D. E. Weinstein, “The Impact of the 2018 Tariffs on Prices and Welfare,” *The Journal of Ecomomic Perspective*, vol. 33, no. 4, pp. 187–210, 2019.

[5] V. Soldatkin and A. Lyrchikova, “Putin says he and Trump should meet to discuss Ukraine and energy prices,” Reuters.

[6] Huaxia, “Kremlin says trade wars negatively affect all countries,” Xinhua.

[7] R. R. Pérez and J. Bajorath, “Evolution of Support Vector Machine and Regression Modeling in Chemoinformatics and Drug Discovery,” *J Comput Aided Mol Des*, vol. 36, no. 5, pp. 355–362, May 2022, doi: 10.1007/s10822-022-00442-9.

[8] A. Goel, A. K. Goel, and A. Kumar, “The role of artificial neural network and machine learning in utilizing spatial information,” *Spatial Information Research*, vol. 31, no. 3, pp. 275–285, Jun. 2023, doi: 10.1007/s41324-022-00494-x.

[9] W. Na, W. Zhang, S. Yan, and G. Liu, “Automated Neural-Based Modeling of Microwave Devices Using Parallel Computation and Interpolation Approaches,” *IEEE Access*, vol. 7, pp. 73929–73937, 2019, doi: 10.1109/ACCESS.2019.2918268.

[10] J. Wang, X. Wang, J. Li, and H. Wang, “A Prediction Model of CNN-TLSTM for USD/CNY Exchange Rate Prediction,” *IEEE Access*, vol. 9, pp. 73346–73354, 2021, doi: 10.1109/ACCESS.2021.3080459.

[11] A. Hamooni, S. Tajdini, M. Qezelbash, F. Jafari, and N. Ebrahimiyan, “Monetary behavior theory in long-term and turbulent conditions on the Russian Ruble,” *Journal of Mathematics and Modeling in Finance*, vol. 2, no. 1, pp. 183–194, 2022, doi: 10.22054/jmmf.2022.14572.

[12] S. Soewignjo, N. W. W. Septia Sari, A. P. Mediani, M. A. Z. Kamil, D. Amelia, and N. Chamidah, "Chinese Yuan Exchange Rate Against the Indonesian Rupiah Prediction Using Support Vector Regression," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 18, no. 3, pp. 1683–1694, 2024, doi: 10.30598/barekengvol18iss3pp1683-1694.

[13] J. Acquah, A. Emmanuel Nti, I. Ampofi, and D. Akorli, "Artificial Neural Network Model for Predicting Exchange Rate in Ghana: A Case of GHS/USD," *American Journal of Mathematical and Computer Modelling*, vol. 7, no. 1, p. 1, 2022, doi: 10.11648/j.ajmcm.20220701.11.

[14] P. L. Makoni, "Foreign Portfolio Investments, Exchange Rates and Capital Openness: A Panel Data Approach," *International Journal of Economics and Business Administration*, vol. 8, no. 2, pp. 100–113, 2020.

[15] H. Hassani, L. Marvian, M. Yarmohammadi, and M. R. Yeganegi, "Unraveling Time Series Dynamics: Evaluating Partial Autocorrelation Function Distribution and Its Implications," *Mathematical and Computational Applications*, vol. 29, no. 4, p. 58, Jul. 2024, doi: 10.3390/mca29040058.

[16] W. W. S. Wei, *Time Series Analysis Univariate and Multivariate Methods*, 2nd editio. Philadelphia: Greg Tobin, 2006.

[17] H. Prabowo, S. Suhartono, and D. D. Prastyo, "The Performance of Ramsey Test, White Test and Terasvirta Test in Detecting Nonlinearity," *Inferensi*, vol. 3, no. 1, p. 1, 2020, doi: 10.12962/j27213862.v3i1.6876.

[18] R. Saputra, C. Agustina, D. Puspitasari, R. Ramanda, Warjiyono, D. Pribadi, Lisnawanty, and K. Indriani, "Detecting Alzheimer's Disease by The Decision Tree Methods Based On Particle Swarm Optimization," *J Phys Conf Ser*, vol. 1641, no. 1, p. 012025, Nov. 2020, doi: 10.1088/1742-6596/1641/1/012025.

[19] E. Khosla, R. Dharavath, and R. Priya, "Crop yield prediction using aggregated rainfall-based modular artificial neural networks and support vector regression," *Environ Dev Sustain*, vol. 22, no. 6, pp. 5687–5708, 2020, doi: 10.1007/s10668-019-00445-x.

[20] W. S. Sarle, "Neural Networks and Statistical Models," *Proceedings of the Nineteenth Annual SAS Users Group International Conference (North Carolina)*, 1994.

[21] E. Pusporani, Suhartono, and D. D. Prastyo, "Hybrid multivariate generalized space-time autoregressive artificial neural network models to forecast air pollution data at Surabaya," 2019, p. 020090. doi: 10.1063/1.5139822.

[22] R. J. Hyndman, "Cross-validation for time series." Accessed: Mar. 07, 2025. [Online]. Available: https://robjhyndman.com/hyndsight/tscv/?form=MG0AV3

[23] M. Bennett, M. Nekouei, A. P. R. Mehta, E. Kleczyk, and K. Hayes, "Methodology to Create Analysis-Naive Holdout Records as well as Train and Test Records for Machine Learning Analyses in Healthcare," no. May, 2022, doi: 10.48550/arXiv.2205.03987.

[24] W. K. M. Mahmood, "Determination of Outdoor Air Quality Level," vol. 9, no. 1, pp. 45–54, 2021.

[25] B.Y.Qu, B. F. Lang, J.J.Liang, A. K. Qin, and O.D.Crisalle, "Two-hidden-layer extreme learning machine for regression and classification," vol. 175(part A, pp. 826–834, 2016, doi: https://doi.org/10.1016/j.neucom.2015.11.009.

[26] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An adaptive SVR for high-frequency stock price forecasting," *IEEE Access*, vol. 6, pp. 11397–11404, 2018, doi: 10.1109/ACCESS.2018.2806180.

[27] R. K. Dash, T. N. Nguyen, and K.Cengiz, "Fine-tuned support vector regression model for stock predictions," *Neural Comput & Applic*, vol. 35, 2023, doi: https://doi.org/10.1007/s00521-021-05842-w.

[28] M. Ali and P. J, "Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements," *ARO-The Scientific Journal of Koya University*, vol. 10, no. 1, pp. 85–91, Jun. 2022, doi: 10.14500/aro.10955.

[29] T. O. Hodson, T. M. Over, and S. S. Foks, "Mean Squared Error, Deconstructed," *J Adv Model Earth Syst*, vol. 13, no. 12, Dec. 2021, doi: 10.1029/2021MS002681.

[30] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci Model Dev*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/gmd-15-5481-2022.

[31] R. G. Silva, M. H. D. M. Ribeiro, S. R. Moreno, V. C. Mariani, and L. S. Coelho, "A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting," *Energy*, vol. 216, p. 119174, Feb. 2021, doi: 10.1016/j.energy.2020.119174.